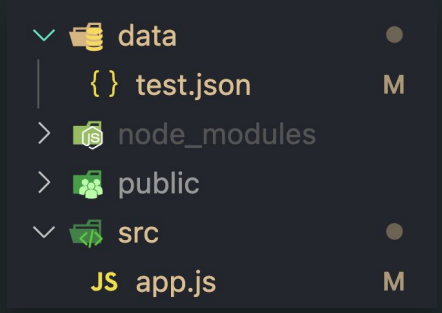
NODE - Path

Path es el módulo de node que nos ayuda a trabajar con las rutas de los directorios y archivos entre otras funciones extra.

Uno de los métodos disponibles que tenemos es resolve que nos va permitir unir varios segmentos de una ruta para formar una ruta final.



Para llegar al archivo test.json desde app nos viene muy bien utilizar resolve.

Con __dirname accedemos a la ruta actual de app y después le decimos que desde el punto donde nos encontramos, la ruta hasta el archivo es esa, y resolve construirá la ruta correctamente

```
const testFile = path.resolve(__dirname, '../data/test.json');
```

NODE - File System - callbacks

File System es el módulo de node que nos permite leer, escribir, copiar, mover, eliminar... archivos en nuestro servidor.

```
const fs = require('fs');
```

De momento vamos a trabajar con el método readFile, que sirve para leer archivos.

```
fs.readFile(file, (err, data) => {
  //código a ejecutar
})
```

file: Es el archivo que vamos a leer.
callback(err, data).
err: Almacena los errores al leer el archivo.
data: Es la información del archivo leído.

NODE - File System - callbacks

Ejemplo de cómo leer los datos de un JSON y enviarlos al cliente en formato callback.

```
app.get('/test', (req, res) => {

  fs.readFile('./data/test.json', (err, data) => {

    if (err) res.status(500).send('Error al leer el archivo');

    const jsonData = JSON.parse(data);

    res.send(jsonData);
  });
});
```

NODE - File System - promesas

Ejemplo de cómo leer los datos de un JSON y enviarlos al cliente en formato promesa.

```
app.get('/test', async (req, res) => {

  try {

    const data = await fsPromise.readFile('./data/test.json');

    const jsonData = await JSON.parse(data);

    res.send(jsonData);

  } catch (err) {

    res.status(500).send('Error al leer el archivo');
  }
});
```

NODE - File System - write

Ejemplo de cómo escribir datos en un JSON en formato callback

```
app.get('/write', (req, res) => {

  const newInfo = { number: 34 };

  fs.writeFile(testFile, JSON.stringify(newInfo), err => {

    if (err) return res.status(500).send('Error al guardar el archivo');

    res.end();
  });
});
```

NODE - File System - write

Ejemplo de cómo escribir datos en un JSON en formato promesa

```
app.get('/write', (req, res) => {

  const newInfo = { number: 34 };

  try {

    fs.writeFile(testFile, JSON.stringify(newInfo));

    res.end();

  } catch (err) {

    res.status(500).send('Error al guardar el archivo');
  }
});
```

NODE - Estructura de una API

Una API se divide en varias partes, vamos a ver de qué se encarga cada parte de la API

Servidor	Es el punto de entrada de la API.
Rutas	Son las rutas que tiene nuestra API.
Controladores	Son las acciones asociadas a las rutas.

NODE - Controlador en una API

Nuestro controlador es un objeto que vamos a ir rellenando con funciones.

```
const controller = {};
```

```
controller.nombreDeLaFunción = (req, res) => {
  // Código a ejecutar
};
```

Crearemos tantas funciones como necesite nuestra API y una vez creadas exportamos el objeto para poder usarlo en otros archivos.

```
module.exports = controller;
```

NODE - Rutas en una API

Las rutas son cada uno de los puntos a los que podemos acceder en nuestra API, cada ruta tendrá una acción asociada.

```
const express = require('express');
const userRoutes = express.Router();
```

Para gestionar las rutas necesitamos el módulo de express y su método enrutador.

```
const controller = require('../controllers/users.controller');
```

Importamos el controlador.

```
// Acción para el acceso por get a la raiz
userRoutes.get('/', controller.nombreDeLaFunción);
```

Asociamos cada ruta a su acción correspondiente.

NODE - Servidor en una API

El servidor es nuestro punto de entrada a la api, en él debemos establecer dónde están las rutas que va a usar y habilitarlas.

```
// Rutas
const userRoutes = require('./routes/users.routes');
```

Importamos las rutas a nuestro archivo principal

```
// Uso de rutas
app.use('/api/users', userRoutes);
```

Establecemos la ruta de entrada a nuestro archivo de rutas.